УДК 621.391:519.7

АЛГЕБРАЇЧНА АТАКА НА ДВІЙКОВІ SNOW2.0-ПОДІБНІ ПОТОКОВІ ШИФРИ

М.А. Овчарова[1]

[1] *Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Фізико-технічний інститут*

Анотація

У статті викладено основні поняття, які потрібні для розуміння роботи потокового шифру SNOW2.0 та SNOW2.0-подібних потокових шифрів, а також атака на спрощену версію шифру SNOW2.0 та її аналіз, що дозволяє розширити область застосування алгебраїчної атаки на двійкові SNOW2.0-подібні потокові шифри.

Ключові слова: потокові шифри, SNOW2.0, алгебраїчні атаки, стійкість потокових шифрів.

Вступ

У сучасному світі існує достатньо потокових шифрів, вразливості яких ще не знайдені, проте деякі з них відрізняються досить складними перетвореннями. Постає питання, чи можна якось спростити ці шифри, не втрачаючи їх стійкість та не набуваючи нових уразливостей.

Метою дослідження є формулювання умов, які зумовлюють уразливість двійкових SNOW 2.0-подібних потокових шифрів до відомої алгебраїчної атаки. Для досягнення мети спочатку розглянемо атаку на конкретному прикладі.

У цій роботі буде проводитися дослідження процесу перетворення інформації у двійкових SNOW 2.0-подібних шифрах та властивостей компонент алгоритмів шифрування, що визначають їх стійкість відносно алгебраїчних атак. Декілька умовних позначень:

$\oplus$ – операція побітового додавання (за модулем 2).
$\boxplus$ – операція додавання за модулем 2^{32}.
РЗЛЗЗ – регістр зсуву з лінійним зворотним зв'язком.
СА – скінченний автомат.
Rijndael (Advanced Encryption Standard, AES [4]) – симетричний алгоритм блокового шифрування (розмір блока 128 біт, ключ 128/192/256 біт).
S-блок – нелінійна байтова таблиця заміни.

1. Опис SNOW 2.0-подібних шифрів

Потоковий шифр SNOW 2.0 складається з РЗЛЗЗ з 32 бітами та СА з двома 32-бітовими регістрами пам'яті [1]. Схема роботи шифру представлена на рис. 1.

СА призначен для породження нелінійності, з цією метою він реалізує нелінійну біекцію між регістрами пам'яті $\mathcal{S}$, що базується на S-блоці блокового шифру Rijndael.

Правило роботи СА описується такою системою співвідношень:

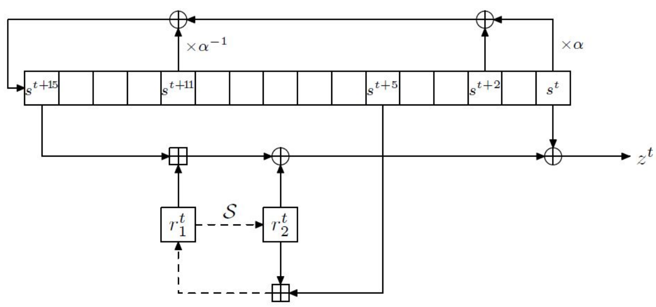

Рис. 1. Схема роботи шифру SNOW 2.0

$$\begin{cases} r_2^{t+1} = \mathcal{S}(r_1^t), \\ r_1^{t+1} = r_2^t \boxplus s^{t+5}, \\ F^t = (r_1^t \boxplus s^{t+15}) \oplus r_2^t, \end{cases}$$

У нормальному режимі вихідний потік гами визначається, як $z^t = s^t \oplus F^t$, або:

$$z^t = s^t \oplus (r_1^t \boxplus s^{t+15}) \oplus r_2^t.$$

Спеціальний режим роботи (ініціалізація шифру, заповнення початкового стану регістру та СА; потік гами при цьому не виробляється) описується трохи іншим співвідношенням:

$$s^{t+16} = \alpha^{-1} s^{t+11} \oplus s^{t+2} \oplus \alpha s^t \oplus F^t.$$

У спрощеній версії шифру SNOW 2.0 модульне додавання $\boxplus$ замінюється на $\oplus$ в його описі, все інше залишається ідентичним [2].

У випадку спрощеної схеми правило роботи СА описується такою системою співвідношень:

$$\begin{cases} r_2^{t+1} = \mathcal{S}(r_1^t), \\ r_1^{t+1} = r_2^t \oplus s^{t+5}, \\ F^t = r_1^t \oplus s^{t+15} \oplus r_2^t, \end{cases}$$

Тепер у нормальному режимі вихідний потік гами визначається як $z^t = s^t \oplus F^t$, або

$$z^t = s^t \oplus (r_1^t \oplus s^{t+15}) \oplus r_2^t,$$

а спеціальний режим роботи – як
$$s^{t+16} = \alpha^{-1} s^{t+11} \oplus s^{t+2} \oplus \alpha s^{t} \oplus F^{t}.$$

S-блок є перестановкою на множині 32-бітових векторів, що базується на раундовій функції шифру Rijndael. Нехай $w = (w_3, w_2, w_1, w_0)$ є вхідними даними S-блоку, де w_i, $i = 0..3$, – чотири байти з w та w_3 – найбільш значущий байт. Нехай

$$w = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{pmatrix}$$

є вектором, що будемо подавати на вхід до S-блоку. Спочатку застосовуємо S-блок до кожного байту вектора, отримуємо:

$$\begin{pmatrix} S_R[w_0] \\ S_R[w_1] \\ S_R[w_2] \\ S_R[w_3] \end{pmatrix}$$

У перетворенні $MixColumn$ раундової функції шифру Rijndael кожен байт розглядається як елемент скінченного поля $\mathbb{F}_{2^8}$, породженого незвідним поліномом
$$x^8 + x^4 + x^3 + x + 1 \in \mathbb{F}_2[x].$$
Відповідно, кожен 4-байтовий вектор може бути представлений поліномом не більше ніж 3-го степеня над $\mathbb{F}_{2^8}$, коефіцієнтами якого виступають байти-координати даного слова. Під час виконання $MixColumn$ поліном, який представляє вхідний вектор, множиться на фіксований поліном
$$c(y) = (x+1)y^3 + y^2 + y + x \in \mathbb{F}_{2^8}[y]$$
за модулем $y^4 + 1 \in \mathbb{F}_{2^8}[y]$; координати результуючого поліному (також степеня не вище 3) утворюють вектор, що є вихідним значенням процедури.

Описане перетворення векторів через множення на поліном (як у шифрі Rijndael) може бути обчислено еквівалентним чином через множення матриць:

$$\begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{pmatrix} = \begin{pmatrix} x & x+1 & 1 & 1 \\ 1 & x & x+1 & 1 \\ 1 & 1 & x & x+1 \\ x+1 & 1 & 1 & x \end{pmatrix} \begin{pmatrix} S_R[w_0] \\ S_R[w_1] \\ S_R[w_2] \\ S_R[w_3] \end{pmatrix}$$

де (r_3, r_2, r_1, r_0) – вектор вихідних байтів S-блоку. Ці байти конкатенуються та формують вихідне слово з S-блоку $r = \mathcal{S}(w)$ [3, 4].

У оригінальному SNOW 2.0 та у його спрощеній версії нелінійність у СА досягається за допомогою описаного відображення між регістрами пам'яті $\mathcal{S}$, що заснована на S-блоці блокового шифру AES. У випадку двійкових SNOW2.0-подібних шифрів, використовуючи заміну модульного додавання $\boxplus$ на $\oplus$, як у спрощеній версії, ми відходимо від S-блоку шифру AES та розглядаємо будь-яку функцію $\mathcal{N}$, що забезпечує нелінійність (рис. 2).

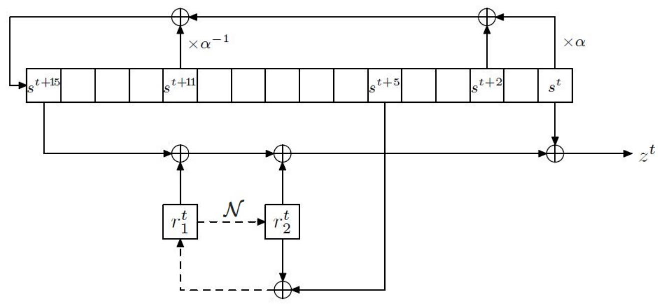

Рис. 2. Схема роботи двійкового SNOW 2.0-подібного шифру

2. Опис алгебраїчної атаки на двійковий SNOW 2.0-подібний шифр

Після опису спрощеної версії шифру SNOW 2.0 у першому розділі перейдемо до самої атаки. Розглянемо генерування ключового потоку та правило оновлення регістру r_1, яке виключає пам'ять із набору рівнянь (рис. 3).

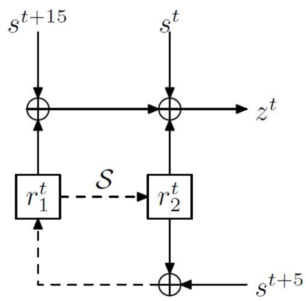

Рис. 3. Схема СА спрощеної версії шифру SNOW 2.0

Отримуємо систему рівнянь
$$\begin{cases} z^t = s^{t+15} \oplus r_1^t \oplus s^t \oplus r_2^t, \\ r_1^t = r_2^{t-1} \oplus s^{t+4}, \end{cases}$$
яка, після підстановки другого рівняння у перше, перетворюєтся на рівняння
$$r_2^t = r_2^{t-1} \oplus z^t \oplus s^{t+15} \oplus s^{t+4} \oplus s^t.$$

З цього рівняння отримуємо вираз для регістра r_2 для будь-якого кроку t, що включає елементи ключового потоку, початкове заповнення РЗЛЗЗ s^0, ..., s^{15} та початкове заповнення r_2^0 регістру r_2. Заносимо до рівняння для кожного кроку t відомі коефіцієнти e_t^i:
$$r_2^t = r_2^0 \bigoplus_{i=0}^{t} z_i \bigoplus_{j=0}^{15} e_t^j s_j.$$

Нехай $t = 0$ для першої вихідної послідовності гами. Легко перевірити, що регістр r_1 оновлюється за правилом $r_1^{t+1} = r_2^t \oplus s^{t+5}$ (також початковий стан регістру r_1 можна отримати зі стану r_2^0 та зі співвідношення $r_1^0 = r_2^0 \oplus s^0 \oplus s^{15} \oplus z^0$). Іншими словами, ми позбулися впливу пам'яті на кроки $t > 0$, оскільки побудували лінійну залежність із початковим заповненням РЗЛЗЗ та r_2^0.

На рис. 4 наведена ілюстрація лінійної залежності між $r_2^0, s^0, ..., s^{15}$ у регістрах пам'яті r_1 та r_2.

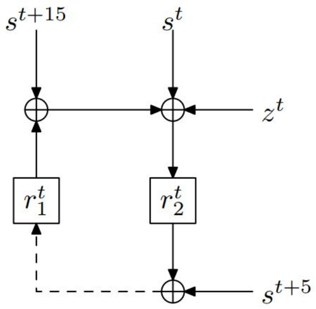

Рис. 4. Відстеження регістрів пам'яті r_1 та r_2

З огляду на те, що функція $\mathcal{S}$ побудована на S-блоці шифру Rijndael, отримуємо систему з 156-ти квадратичних рівнянь, яка описує залежність між r_1^t та r_2^{t+1} [5].

Відновлення початкового стану РЗЛЗЗ та ключа можна звести до знаходження розв'яку такої системи, проте існує дві різні стратегії.

Перша базується на лінеарізації системи, проте у такому випадку вона потребує приблизно 2^{51} операцій.

Інша стратегія полягає у пошуку розв'язку систем квадратичних рівнянь без лінеризації. У такому випадку складність рішення системи значно вища за вирішення лінеарізованої.

Як тільки початковий стан $s^0, ..., s^{15}$ та r_2^0 буде відновлений, використовуючи метод лінеарізації, r_1^0 буде отримано з виразу $r_1^0 = r_2^0 \oplus s^0 \oplus s^{15} \oplus z^0$, так що стане відомо повний стан шифру при $t = 0$. Для того, щоб отримати секретний ключ K (і, таким чином, передбачувати в подальшому послідовність гами для інших початкових станів), достатньо запустити один раз шифр у зворотньому порядку у нормальному режимі та 32 рази у спеціальному режимі зворотнього зв'язку. З цього випливає, що переходи станів SNOW 2.0 у нормальному та у спеціальному режимах є оберненими. Зважаючи на це, ми зможемо отримати стан РЗЛЗЗ під час ініціалізації, а саме значення початкового стану та секретного ключа [2].

3. Розширення алгебраїчної атаки на двійкові SNOW 2.0-подібні шифри

У попередньому розділі було наведено приклад атаки на спрощену версію потокового шифру SNOW2.0, що дозволяє виділити основні кроки її побудови, дію яких можна перенести на двійкові SNOW2.0-подібні шифри.

Спочатку дамо визначення алгебраїчної імунності S-блоку. Розглянемо деякий S-блок як векторну булеву функцію $S : V_n \to V_n$ з координатними функціями $s_1, ..., s_n$ та ідеал кільця булевих функцій від $2n$ змінних $x = (x_1, ..., x_n)$, $y = (y_1, ..., y_n)$, який визначається за формулою

$$I(S) = \langle y_1 \oplus s_1(x), ..., y_n \oplus s_n(x) \rangle .$$

Алгебраїчна імунність S-блоку визначаєтьмя рівністю [6] $AI(S) = \min \deg I(S)$, де

$$\min \deg I(S) = \min\{\deg f : f \in I \setminus \{0\}\}.$$

Загальна схема атаки виглядає таким чином.

Вхід: S-блок, загальна система рівнянь генерації гами;

1) побудова системи булевих рівнянь S-блоку;
2) визначення алгебраїчної імунності S-блоку;
3) визначення системи незалежних рівнянь мінімального степеня між регістрами пам'яті.

Вихід: система рівнянь генерації гами (об'єднання загальної системи рівнянь генерації гами та рівнянь, що описують залежності між регістрами пам'яті).

Саме ця система дає змогу визначити складність провадживаної алгебраїчної атаки, а її розв'язок дозволить відновити початковий стан шифру.

Висновки

У данній статті було розглянуто схему роботи потокового шифру SNOW 2.0 та його модифікацію, наведено приклад атаки на спрощену версію потокового шифру SNOW 2.0, а також описана схема поширення цієї атаки на будь-які двійкові SNOW 2.0-подібні шифри.

Перелік використаних джерел

1. Ekdahl, P., Johansson, T.: A new version of the stream cipher SNOW. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 47–61. Springer, Heidelberg (2003)
2. Billet O., Gilbert H. (2005) Resistance of SNOW 2.0 Against Algebraic Attacks. In: Menezes A. (eds) Topics in Cryptology – CT-RSA 2005. CT-RSA 2005. Lecture Notes in Computer Science, vol 3376. Springer, Berlin, Heidelberg
3. Nyberg K. (1991) Perfect nonlinear S-boxes. In: Davies D.W. (eds) Advances in Cryptology — EUROCRYPT '91. EUROCRYPT 1991. Lecture Notes in Computer Science, vol 547. Springer, Berlin, Heidelberg
4. Daemen J., Rijmen V.: The design of Rijndael, Springer Verlag Series on Information Security and Cryptography, Springer Verlag, 2002, ISBN 3-540-42580-2.
5. Courtois N. T., Pieprzyk J. Cryptanalysis of block ciphers with overdefined systems of equations. In Y. Zheng, editor, Advances in Cryptology- ASIACRYPT 2002, volume 2501 of Lecture Notes in Computer Science, pages 267–287. Springer-Verlag, 2002.
6. Олексійчук А. М. Алгебраїчна імунність векторних булевих функцій та булеві базиси Грьобнера [рукопис].